

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-350650

(P2001-350650A)

(43) 公開日 平成13年12月21日 (2001. 12. 21)

(51) Int.Cl.<sup>7</sup>

G 0 6 F 11/28

識別記号

3 4 0

F I

G 0 6 F 11/28

テーマコード\* (参考)

3 4 0 A 5 B 0 4 2

審査請求 有 請求項の数 9 O L (全 8 頁)

(21) 出願番号 特願2000-172316(P2000-172316)

(22) 出願日 平成12年6月8日 (2000. 6. 8)

(71) 出願人 397065480

エヌ・ティ・ティ・コムウェア株式会社  
東京都港区港南一丁目9番1号

(72) 発明者 麓 裕明

東京都港区港南一丁目9番1号 エヌ・ティ・ティ・コミュニケーションウェア株式会社内

(72) 発明者 向谷 寿徳

東京都港区港南一丁目9番1号 エヌ・ティ・ティ・コミュニケーションウェア株式会社内

(74) 代理人 100064908

弁理士 志賀 正武 (外2名)

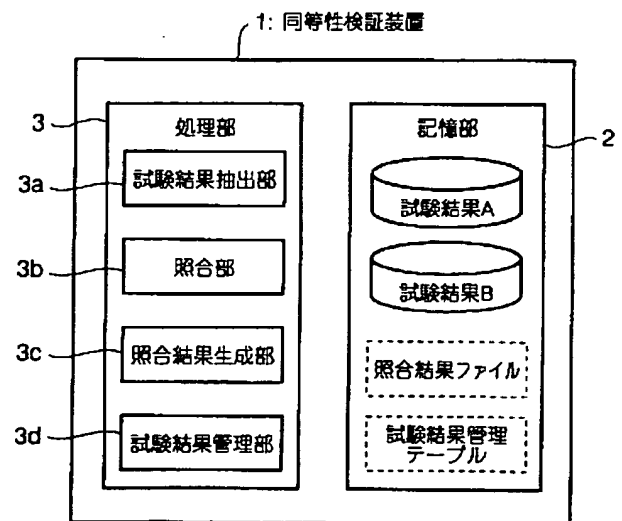
最終頁に続く

(54) 【発明の名称】 ソフトウェアプログラム同等性検証方法及びソフトウェアプログラム同等性検証装置並びにその記録媒体

(57) 【要約】

【課題】 本発明の目的は、ソフトウェアプログラムの既存機能の確認（同等性の検証）を自動的に行え、試験のための作業（人的作業を含む）の削減と品質の向上を実現するソフトウェアプログラム同等性検証方法及び装置を提供することにある。

【解決手段】 本発明のソフトウェアプログラム同等性検証方法は、コンピュータシステム上で、変更前のソフトウェアプログラムと変更後のソフトウェアプログラムとを、同一の試験条件のもとで実行させた試験結果から、同等性の検証に必要な所定の情報を抽出する試験結果抽出段階と、試験結果抽出段階で得られる変更前のソフトウェアプログラムの試験結果から抽出した抽出情報を基準として、所定の試験項目毎に、変更後のソフトウェアプログラムの試験結果から抽出した抽出情報と照合する照合段階を含む。



## 【特許請求の範囲】

【請求項1】 コンピュータシステム上で、変更前のソフトウェアプログラムと変更後のソフトウェアプログラムの同等性を検証するソフトウェアプログラム同等性検証方法において、

前記変更前のソフトウェアプログラムと変更後のソフトウェアプログラムとを、同一の試験条件のもとで実行させた試験結果から、同等性の検証に必要な所定の情報を抽出する試験結果抽出段階と、

前記試験結果抽出段階で得られる前記変更前のソフトウェアプログラムの試験結果から抽出した抽出情報を基準として、所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報と照合する照合段階と、を含むことを特徴とするソフトウェアプログラム同等性検証方法。

【請求項2】 前記照合段階は、

前記所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報が、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報に対し、所定の許容範囲内にある場合、同等性有り（OK）の判定を行い、所定の許容範囲内でない場合、同等性無し（NG）の判定を行う段階を含むことを特徴とする請求項1に記載のソフトウェアプログラム同等性検証方法。

【請求項3】 前記ソフトウェアプログラム同等性検証方法は、

前記照合段階における判定の結果の情報を含む照合結果ファイルを、前記所定の試験項目毎に生成する照合結果生成段階を、さらに含むことを特徴とする請求項2に記載のソフトウェアプログラム同等性検証方法。

【請求項4】 前記ソフトウェアプログラム同等性検証方法は、

前記照合結果ファイルをもとに、全試験項目の試験結果の良否（OK/NG）を含む情報を、前記所定の試験項目毎に記録した試験結果管理テーブルを生成する試験結果管理段階を、さらに含むことを特徴とする請求項3に記載のソフトウェアプログラム同等性検証方法。

【請求項5】 変更前のソフトウェアプログラムと変更後のソフトウェアプログラムの同等性を検証するソフトウェアプログラム同等性検証装置において、

前記変更前のソフトウェアプログラムと変更後のソフトウェアプログラムとを、同一の試験条件のもとで実行させた試験結果から、同等性の検証に必要な所定の情報を抽出する試験結果抽出手段と、

前記試験結果抽出手段で得られる前記変更前のソフトウェアプログラムの試験結果から抽出した抽出情報を基準として、所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報と照合する照合手段と、を具備することを特徴とするソフトウェアプログラム同等性検証装置。

【請求項6】 前記照合手段は、

前記所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報が、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報に対し、所定の許容範囲内にある場合、同等性有り（OK）の判定を行い、所定の許容範囲内でない場合、同等性無し（NG）の判定を、さらに行うことを特徴とする請求項5に記載のソフトウェアプログラム同等性検証方法。

10 【請求項7】 前記ソフトウェアプログラム同等性検証装置は、

前記照合手段における判定の結果の情報を含む照合結果ファイルを、前記所定の試験項目毎に生成する照合結果生成手段を、さらに含むことを特徴とする請求項6に記載のソフトウェアプログラム同等性検証方法。

【請求項8】 前記ソフトウェアプログラム同等性検証装置は、

前記照合結果ファイルをもとに、全試験項目の試験結果の良否（OK/NG）を含む情報を、前記所定の試験項目毎に記録した試験結果管理テーブルを生成する試験結果管理手段を、さらに含むことを特徴とする請求項7に記載のソフトウェアプログラム同等性検証方法。

【請求項9】 コンピュータ装置にインストールすることにより、その装置が請求項1ないし4のいずれかに記載の方法を実行する装置となるソフトウェアが記録されたコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【0001】

30 【発明の属する技術分野】本発明は、ソフトウェアプログラムの修正・変更を行う場合に、当該ソフトウェアプログラムの変更前と変更後の機能の同等性を検証する方法及び装置に関する。

【0002】

40 【従来の技術】近年、ソフトウェアプログラムを組み込んだ様々なシステムが利用されるようになっている。こうしたシステムに対し、機能追加の要望は無いが今後のメンテナンス性を考慮してプログラムが変更される場合や、リグレーションテスト（退行試験）のように機能追加によりプログラムを変更したが、変更個所以外に影響を与えていないことを確認するために、必ず既存機能の確認試験が必要となる。

50 【0003】一方、大規模システム（電話交換機等）では、長い期間にわたり、システムの保守・改善・機能追加等が繰り返行われる。また、小規模であってもオペレーティングシステムやアプリケーションプログラムは、過去に作成された資源をもとにバージョンアップがなされることが多い。このため、システムに組み込まれたプログラムの設計手法およびソフトウェアの構造が、作成時期や作成者により異なる場合があり、また、機能追加があった場合、プログラムがさらに複雑なものとな

っている。これらのことから、現在では、プログラムを整理することにより、開発支援環境を統一し、設計・試験時間を短縮することを目的にソフトウェア構造の見直しが行われるようになっている。

【0004】上記のように、システムに組み込まれたソフトウェアプログラムの修正や機能追加が行われる場合、既存機能の確認試験が行われる。従来は、過去に実施した試験手順書と試験用データをもとに試験を行い、この試験結果と過去の試験結果を比較することにより、ソフトウェアプログラムの機能の正常性の確認を実施していた(図3(a)参照)。

【0005】

【発明が解決しようとする課題】しかし、このような方法では、過去の試験結果を管理する必要がある、また試験結果の確認には詳細な要求仕様の把握が必要となり、かなりの時間を要していた。

【0006】本発明は、上記の点に鑑みてなされたもので、ソフトウェアプログラムの既存機能の確認(同等性の検証)を自動的に行え、試験のための作業(人的作業を含む)の削減と品質の向上を実現するソフトウェアプログラム同等性検証方法及び装置を提供するものである。

【0007】

【課題を解決するための手段】本発明のソフトウェアプログラム同等性検証方法は、コンピュータシステム上で、変更前のソフトウェアプログラムと変更後のソフトウェアプログラムの同等性を検証するソフトウェアプログラム同等性検証方法において、前記変更前のソフトウェアプログラムと変更後のソフトウェアプログラムとを、同一の試験条件のもとで実行させた試験結果から、同等性の検証に必要な所定の情報を抽出する試験結果抽出段階と、前記試験結果抽出段階で得られる前記変更前のソフトウェアプログラムの試験結果から抽出した抽出情報を基準として、所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報と照合する照合段階と、を含むことを特徴とする。

【0008】また、本発明のソフトウェアプログラム同等性検証方法において、前記照合段階は、前記所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報が、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報に対し、所定の許容範囲内にある場合、同等性有り(OK)の判定を行い、所定の許容範囲内にない場合、同等性無し(NG)の判定を行う段階を含むことを特徴とする。

【0009】また、本発明のソフトウェアプログラム同等性検証方法において、前記照合段階における判定の結果の情報を含む照合結果ファイルを、前記所定の試験項目毎に生成する照合結果生成段階を、さらに含むことを特徴とする。

【0010】前記ソフトウェアプログラム同等性検証方

法は、前記照合結果ファイルをもとに、全試験項目の試験結果の良否(OK/NG)を含む情報を、前記所定の試験項目毎に記録した試験結果管理テーブルを生成する試験結果管理段階を、さらに含むことを特徴とする。

【0011】本発明のソフトウェアプログラム同等性検証装置は、変更前のソフトウェアプログラムと変更後のソフトウェアプログラムの同等性を検証するソフトウェアプログラム同等性検証装置において、前記変更前のソフトウェアプログラムと変更後のソフトウェアプログラムとを、同一の試験条件のもとで実行させた試験結果から、同等性の検証に必要な所定の情報を抽出する試験結果抽出手段と、前記試験結果抽出手段で得られる前記変更前のソフトウェアプログラムの試験結果から抽出した抽出情報を基準として、所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報と照合する照合手段と、を具備することを特徴とする。

【0012】また、本発明のソフトウェアプログラム同等性検証装置において、前記照合手段は、前記所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報が、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報に対し、所定の許容範囲内にある場合、同等性有り(OK)の判定を行い、所定の許容範囲内にない場合、同等性無し(NG)の判定を、さらに行うことを特徴とする。

【0013】また、本発明のソフトウェアプログラム同等性検証装置は、前記照合手段における判定の結果の情報を含む照合結果ファイルを、前記所定の試験項目毎に生成する照合結果生成手段を、さらに含むことを特徴とする。

【0014】また、本発明のソフトウェアプログラム同等性検証装置は、前記照合結果ファイルをもとに、全試験項目の試験結果の良否(OK/NG)を含む情報を、前記所定の試験項目毎に記録した試験結果管理テーブルを生成する試験結果管理手段を、さらに含むことを特徴とする。

【0015】また、本発明は、コンピュータ装置にインストールすることにより、その装置が請求項1ないし4のいずれかに記載の方法を実行する装置となるソフトウェアが記録されたコンピュータ読取可能な記録媒体である。

【0016】

【発明の実施の形態】以下、本発明の実施の形態を、図面を参照して説明する。図1は、本発明のソフトウェアプログラム同等性検証方法の一実施の形態である手順からなるソフトウェアプログラムを、コンピュータシステムにインストールした装置、あるいは、本発明のソフトウェアプログラム同等性検証装置の一実施の形態である同等性検証装置1の構成を示すブロック図である。

【0017】まず、はじめに本発明のソフトウェアプロ

グラム同等性検証方法の概要を説明する。本発明のソフトウェアプログラム同等性検証方法は、ソフトウェアプログラムの変更や機能追加を行う場合に、既存の機能（無変更の機能）には変更が無いことから、変更前後のソフトウェアプログラムに対し同じ条件で試験を行えば、同じ結果が得られることを利用している。すなわち、図2に示すように、変更前後のソフトウェアプログラムに同じ試験条件Aを入力すれば、変更前のソフトウェアプログラムの試験結果Aと変更後のソフトウェアプログラムの試験結果Bは同じ結果となることを利用し、試験結果Aを基準値として、試験結果Bとの照合を行い、相違が許容範囲にあるか否かを判定し、ソフトウェアプログラムの同等性の検証を行うものである。

【0018】図3に、同等性検証装置1において実施される、変更前および変更後のソフトウェアプログラムの同等性の検証の方法（図3（b））と、従来の方法（図3（a））との対比を示している。同等性検証装置1では、変更前後のソフトウェアプログラムにたいし同じ条件で試験を行うため、過去の試験結果を管理する必要が無く、また試験結果を自動的に照合することにより、詳細な要求仕様の把握が不要となる。なお、変更前および変更後のソフトウェアプログラムの同等性の検証するにあたり、変更前後のソフトウェアプログラムで試験を行うための試験項目数が従来の方法に比べ倍になるが、これについては自動試験が可能であり、大きな作業の負荷はかからない。

【0019】以下に説明する本実施の形態では、現実の試験結果A、Bに、システムからのメッセージの他に、各種試験機器、試験環境（ワークステーション）等からの情報（応答メッセージ等）が含まれているため、基準となる試験結果Aと、試験結果Bを単純に比較するのが困難となる。そこで別途定められたキーワードにより照合に必要なデータを抽出し、この抽出したデータを使って、実際の照合を行っている（図4参照）。

【0020】ここで、同等性検証装置1の構成を説明する。同等性検証装置1は、変更前ソフトウェアプログラムの試験結果Aと変更後ソフトウェアプログラムの試験結果Bを記憶する記憶部2と、下記の試験結果抽出部3aと、照合部3bと、照合結果生成部3cと、試験結果管理部3dとからなる処理部3とから構成される。

【0021】試験結果抽出部3a：変更前ソフトウェアプログラムの試験結果Aと、変更後ソフトウェアプログラムの試験結果Bから同等性の検証に必要な所定の情報を、別途定められるキーワードに基づき抽出し、抽出した情報からなるファイルを生成する。

照合部3b：試験結果抽出部3aにより試験結果Aから抽出された抽出情報A'を基準として、同様に試験結果Bから抽出された抽出情報B'を照合する。このとき、別途定められたチェックポイントについて照合を行う。そして抽出情報B'が、基準となる抽出情報A'から所

定の許容範囲内にあるか否かを判断し、所定の許容範囲内にある場合、同等性有り（OK）の判定を行い、所定の許容範囲内でない場合、同等性無し（NG）の判定を行う。

【0022】照合結果生成部3c：照合部3bによる判定の結果の情報を含む照合結果ファイルを、試験項目毎に生成する。

試験結果管理部3d：照合結果生成部3cが生成した照合結果ファイルをもとに、全試験項目の試験結果の良否（OK/NG）を含む情報を、試験項目毎に記録した試験結果管理テーブルを生成する。

【0023】なお、上記試験結果Aと試験結果Bは、別途定義された試験項目に基づく試験条件のもと、ターゲットのシステム上で、変更前ソフトウェアプログラムと変更後ソフトウェアプログラムのそれぞれを実行させ試験した結果である。また、所定の許容範囲とは、当然のごとくシステムにより異なるものであり、試験するシステム毎に適宜定義されるものである。

【0024】また、同等性検証装置1において、記憶部2は、ハードディスク、光磁気ディスク等の不揮発性の記録装置により構成されている。また、処理部3はメモリおよびCPU（中央演算装置）等により構成され、処理部3の各機能を実現するためのソフトウェアプログラム（図示せず）をメモリにロードして実行することによりその機能が実現されるものとする。また、同等性検証装置1には、周辺機器として入力装置、表示装置等（いずれも図示せず）が接続されるものとする。ここで、入力装置とはキーボード、マウス等の入力デバイスのことをいう。表示装置とはCRT（Cathode Ray Tube）や液晶表示装置等のことをいう。

【0025】次に、このように構成された同等性検証装置1の動作について説明する。

【0026】はじめに、試験者により、所定の試験手順書に基づき、ターゲットのシステム上で所定の試験用データを使い、変更前のソフトウェアプログラムと変更後のソフトウェアプログラムが実行され、変更前のソフトウェアプログラムの試験結果Aと変更後のソフトウェアプログラム試験結果Bを得て、同等性検証装置1の記憶部2に記憶されているものとする。

【0027】以下の説明では、特に電話交換機をターゲットとしたものとしている。この場合、上記試験結果のファイルの一例としては、試験結果ファイル、トラヒックデータファイル、信号モニタ解析ファイルがあげられる。図5～7に、上記各ファイルの内容例を示している。

【0028】はじめに、試験結果抽出部3aは、試験結果ファイルから、「試験開始」、「正常終了」等のキーワードに基づき同等性の検証に必要な所定の情報を抽出し、これらの情報からなる結果ファイルを生成する（図9参照）。同様に、トラヒックデータファイルから、

「運用情報」等をキーワードとして、また信号モニタ解析ファイルから、「MSGTYPE」等をキーワードとして、それぞれに対し所定の情報からなるデータファイルおよびモニタファイルを生成する（図8：ステップS1）。

【0029】次に、照合部3bは、ステップS1において、キーワードに基づき所定の情報を抽出した情報からなる結果ファイル、データファイル、モニタファイルから、図10に示すチェックポイントについて照合を行い、同等性有り／無しの判定を行う（図8：ステップS2）。

【0030】図10に示した例では、変更前／変更後のソフトウェアプログラムの試験結果から得た2つの結果ファイルに対するチェックポイントとして、「疑似呼正常発呼」か、「緊急／最緊急メッセージ出力無し」か、「各種リソースの浮き上がり無し」か、を用いてOK／NGの判定を行い、さらに、「自立メッセージ」、「各種データ出力情報」、「呼処理トレース情報」をチェックポイントとして照合・判定を行う。同様に、データファイルに対しては、「運用情報」、「トラヒック情報（不完了呼（正常に接続できなかった呼）の時刻情報と相手局呼識別番号、自局識別番号は対象外）」をチェックポイントとして照合・判定を行う。そしてまた、モニタファイルに対しては、電話交換機が受信する信号内容である「信号モニタ情報（通話路確認信号と相手局呼識別番号、自局識別番号は対象外）」をチェックポイントとして照合・判定を行う。

【0031】なお、チェックポイントの「疑似正常発呼」とは、通常、電話機から発信することにより電話交換機へアクセスされるが、発信する全ての電話機を準備し試験することができないため、電話機に相当する疑似装置により各種条件の信号を作成し交換機へアクセスさせ確認試験を行うことから、正常に疑似装置から発信（発呼）できたことの確認を行うためのチェックポイントである。

【0032】また、チェックポイントの「緊急／最緊急メッセージ出力無し」とは、通常、電話交換機の状態または電話の接続に対して不具合が発生した場合の保守者への通知があり、このメッセージが試験実施時に発生した場合、何らかの原因があると考えられるため、この確認のためのチェックポイントである。また、チェックポイントの「各種リソースの浮き上がり無し」とは、通常、電話交換機が各種メモリ等のリソース管理を行っており、正常にソフトウェアプログラムが実行されると、リソースを捕捉し、処理終了後、捕捉したリソースを解放するが、このリソースの補足ならびに開放が正常になされているかの確認を行うためのチェックポイントである。

【0033】また、チェックポイントの「自立メッセージ」とは、緊急／最緊急メッセージの他に、通常のメッ

セージとして保守者へ通知されるものがあり、例えば、あるコマンドを時間指定登録しておけば、該当時間になると登録コマンドの実行を通知するメッセージのことである。また、チェックポイントの「呼処理トレース情報」とは、電話接続の正常性を確認するためのコマンド（呼処理トレースコマンド）に係る情報のことである。この「呼処理トレース情報」には、電話接続の正常性を確認するためのコマンドに対する電話交換機からの応答であるイベント番号（EVENO）や、接続先番号、発信者番号等が含まれる。

【0034】次に、照合結果生成部3cは、試験項目毎に、照合部3bによる照合・判定の結果を照合結果を含めたファイル（照合結果ファイル）を生成する（図8：ステップS3）。図11に、照合OK時（同等性有り）と、照合NG時（同等性無し）の場合の、照合結果ファイルの一例をそれぞれ図示している。これらのファイルには、試験項目番号のほか、トレース情報、緊急メッセージ情報、モニタ情報の照合結果ならびに試験者名、判定の結果（OK／NG）、日付等が含まれる。照合NG時には、例えば、トレース情報の照合結果として、電話交換機に投入されたコマンドパラメータ：RTR（ルートトレースの略）に対するエラーメッセージ（ERR-XX発生）が記録される。その他緊急メッセージ情報やモニタ情報が、簡易に識別できるように記録される。

【0035】次に、試験結果管理部3dは、照合結果生成部3cが生成した各照合結果ファイルから、全試験項目の試験結果管理テーブルを生成する（図8：ステップS4）。図12に試験結果管理テーブルの一例を示している。この試験結果管理テーブルは、試験項目番号、試験実施者の氏名、EVENO（上記イベント番号）、OK／NGの別、実施日の項目からなっている。変更後のソフトウェアプログラムの試験進捗管理者は、この試験結果管理テーブルを利用し、進捗管理を行うことができる。

【0036】なお、上記で説明した動作フローは一例であり、上記の処理の流れに限定されるものではない。また、図1における処理部3の機能を実現するためのソフトウェアプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたソフトウェアプログラムをコンピュータシステムに読み込ませ、実行することによりソフトウェアプログラム同等性検証方法を実施してもよい。なお、ここでいう「コンピュータシステム」とは、OSや周辺機器等のハードウェアを含むものとする。

【0037】また、「コンピュータ読み取り可能な記録媒体」とは、フロッピー（登録商標）ディスク、光磁気ディスク、ROM、CD-ROM等の可搬媒体、コンピュータシステムに内蔵されるハードディスク等の記憶装置のことをいう。さらに「コンピュータ読み取り可能な記録媒体」とは、インターネット等のネットワークや電

10

20

30

40

50

話回線等の通信回線を介してソフトウェアプログラムが送信された場合のサーバやクライアントとなるコンピュータシステム内部の揮発性メモリ（RAM）のように、一定時間ソフトウェアプログラムを保持しているものも含むものとする。

【0038】また、上記ソフトウェアプログラムは、このソフトウェアプログラムを記憶装置等に格納したコンピュータシステムから、伝送媒体を介して、あるいは、伝送媒体中の伝送波により他のコンピュータシステムに伝送されてもよい。ここで、ソフトウェアプログラムを伝送する「伝送媒体」は、インターネット等のネットワーク（通信網）や電話回線等の通信回線（通信線）のように情報を伝送する機能を有する媒体のことをいう。また、上記ソフトウェアプログラムは、前述した機能の一部を実現するためのものであっても良い。さらに、前述した機能をコンピュータシステムにすでに記録されているソフトウェアプログラムとの組み合わせで実現できるもの、いわゆる差分ファイル（差分プログラム）であってもよい。

【0039】以上、この発明の実施の形態を、図面を参照して詳述してきたが、具体的な構成はこの実施形態に限られるものではなく、この発明の要旨を逸脱しない範囲の設計等も含まれる。

【0040】

【発明の効果】以上、詳細に説明したように、本発明によれば、コンピュータシステム上で、変更前のソフトウェアプログラムと変更後のソフトウェアプログラムとを、同一の試験条件のもとで実行させた試験結果から、同等性の検証に必要な所定の情報を抽出し、前記試験結果抽出段階で得られる前記変更前のソフトウェアプログラムの試験結果から抽出した抽出情報を基準として、所定の試験項目毎に、前記変更後のソフトウェアプログラムの試験結果から抽出した抽出情報とを照合している。これにより、ソフトウェアプログラムの既存機能の確認

（変更前後のソフトウェアプログラムの同等性の検証）を行うために、過去の試験結果を管理する必要がなくなり、また、試験者は、詳細な要求仕様の把握も必要なくなり、このための時間も必要としない。すなわち、試験結果の自動照合により試験のための作業の削減と品質の向上が実現する。

【図面の簡単な説明】

【図1】 本発明の一実施の形態である同等性検証装置の構成を示すブロック図である。

【図2】 本発明のソフトウェアプログラム同等性検証方法を説明する図である。

【図3】 従来の方と、本発明のソフトウェアプログラム同等性検証方法を対比し示している図である。

【図4】 本発明の一実施の形態である同等性検証装置を説明する図である。

【図5】 試験結果ファイルの一例である。

【図6】 トラフィックデータファイルの一例である。

【図7】 信号モニタ解析ファイルの一例である。

【図8】 本発明の一実施の形態である同等性検証装置の動作フローチャートである。

【図9】 試験結果ファイル毎に、適用されるキーワードと生成される抽出ファイルの一例を示す図である。

【図10】 抽出ファイル毎のチェックポイントの一例を示す図である。

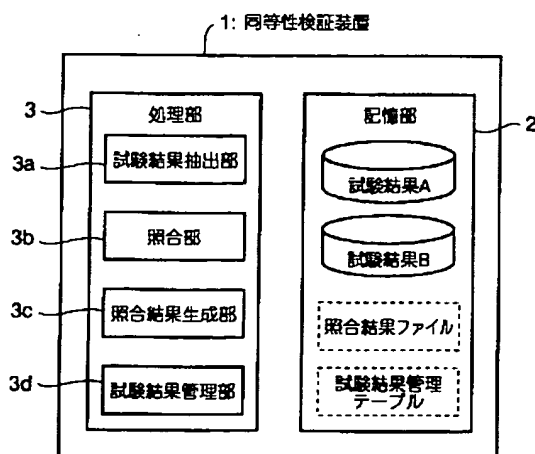
【図11】 照合結果ファイルの一例である。

【図12】 試験結果管理テーブルの一例である。

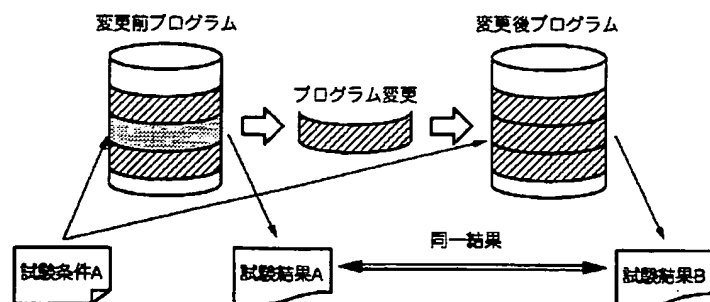
【符号の説明】

1…同等性検証装置	2…記憶部
3…処理部	3a…試験結果抽出部
3b…照合部	3c…照合結果生成部
3d…試験結果管理部	3d…試験結果管理部

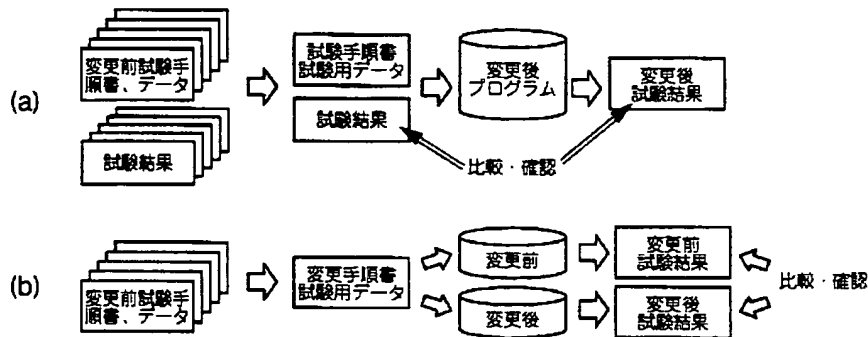
【図1】



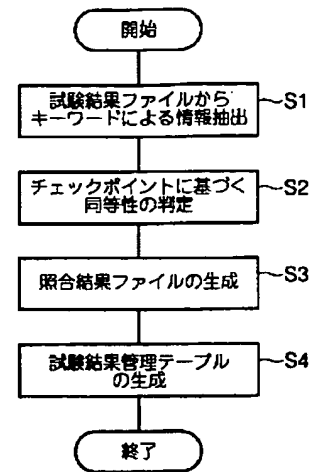
【図2】



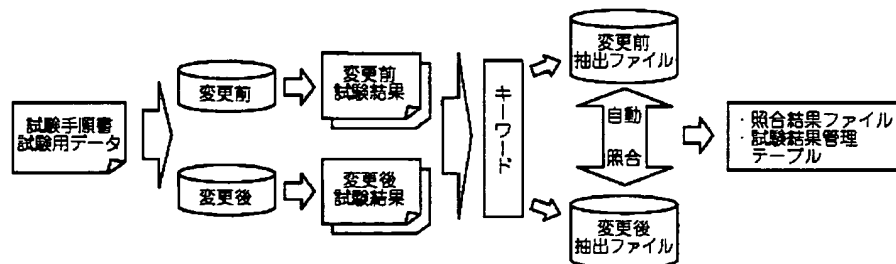
【図3】



【図8】



【図4】



【図5】

試験結果ファイル

```

#### 試験開始: bua210001 #### ← キーワード
¥DTY Y1999.M11.D26.H10.M10
.
#### トラヒック測定 ####
¥DSP TRF/
# DSPTRF #
000000000 000000000 000000000 } チェックポイント
...DSP END 11/26 10:10 (A)
.
共通線信号ファイルを設定した
## 1 call 発呼 ##
CSR CNT=00000000 SI=ANET SSF=00 HiHo=SDRQ CiCo=00 ← チェックポイント
# ..SCP.TRACE START 11/26 10:10 (A) ← チェックポイント
SVCNID:ALP AN:990xxxxx
.
#### TRAFFIC DATA OUT (DBG) ####
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
.
¥DBG SVC OUT/
#### SVC TRF:SVCTFT (DBG) ####
SETUZOKU FUKANOU: 0000000001 UP } チェックポイント
BUSY SOUGUU : 0000000001 UP

```

【図9】

【図6】

トラヒックデータファイル

```

★★★ 運用情報解析結果 (NSP共通) ★★★ ← キーワード
該当の試験では、運用情報の出力がありません
●●● トラヒック情報出力(ALP) Ver2.00 ●●●
世代識別 : [1]
データ識別 : [9]
ユーザ情報幅 : [60]
発信局番号 : [80-02-018]
情報提供事業者サービス : [990xxxxxx]

```

【図7】

信号モニタ解析ファイル

```

MSGTYPE DPC OPC SLS CLS ← キーワード
XUOT 1A5E 000E 00 81
.
データ長 : [37]
ユーザクラス : [1:高度電話網サービス]
相手局呼識別番号 : [000000]
自局呼識別番号 : [201100]
シーケンス番号 : [1]
信号種別 : [11:加入者データ問合せ]

```

試験結果のファイル	キーワード	抽出ファイル
試験結果ファイル	試験開始、正常終了	結果ファイル
トラヒックデータファイル	運用情報	データファイル
信号モニタ解析ファイル	MSGTYPE	モニタファイル

【図10】

抽出ファイル	チェックポイント
結果ファイル	類似呼正常発呼 緊急/最緊急メッセージ出力無し 各種リソースの浮き上がり無し 自立メッセージ照合 各種データ出力情報照合 呼処理トレース情報照合
データファイル	運用情報、トラヒック情報照合 (不完了呼の時刻情報と相手局呼識別番号、 自局識別番号は照合対象外)
モニタファイル	信号モニタ情報照合 (通話路確認信号と相手局呼識別番号、 自局識別番号は照合対象外)

【図11】

照合OK時	照合NG時
>>>> 試験項目番号 >>>> —— トレース情報 —— —— 緊急メッセージ情報 —— —— モニタ情報 —— 試験項目番号 試験者名 OK 日付	>>>> 試験項目番号 >>>> —— トレース情報 —— RTR...ERR-XX発生 —— 緊急メッセージ情報 —— XXXX スタック 収集 NG —— モニタ情報 —— 結 果 xxx 情報 0:必要 期待値 xxx 情報 1:不要 試験項目番号 試験者名 NG 日付

【図12】

試験項目番号	試験実施者	EEV-NO	OK/NG	実施日
0001	担当者名	00000000	OK	0209
0002	担当者名	00000000	OK	0209
0003	担当者名	00000000	OK	0209
0004	担当者名	03130408	NG	0209
;	;	;	;	;

フロントページの続き

(72)発明者 友田 正毅  
東京都港区港南一丁目9番1号 エヌ・テ  
ィ・ティ・コミュニケーションウェア株式  
会社内

(72)発明者 茂木 秀伸  
東京都港区港南一丁目9番1号 エヌ・テ  
ィ・ティ・コミュニケーションウェア株式  
会社内

(72)発明者 黒川 澄  
東京都港区港南一丁目9番1号 エヌ・テ  
ィ・ティ・コミュニケーションウェア株式  
会社内

Fターム(参考) 5B042 HH12



# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-350650

(43)Date of publication of application : 21.12.2001

(51)Int.Cl.

G06F 11/28

(21)Application number : 2000-172316

(71)Applicant : NTT COMWARE CORP

(22)Date of filing : 08.06.2000

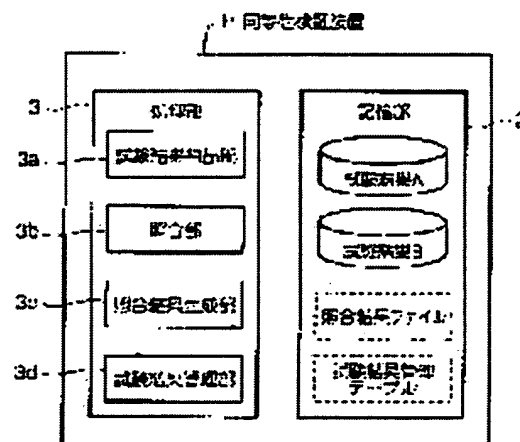
(72)Inventor : FUMOTO HIROAKI  
MUKOYA KAZUNORI  
TOMOTA MASAKI  
MOGI HIDENOBU  
KUROKAWA KIYOSHI

## (54) INSPECTING METHOD AND DEVICE FOR SOFTWARE PROGRAM EQUIVALENCY, AND RECORDING MEDIUM THEREFOR

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide an inspecting method and device for software equivalency capable of automatically performing the confirmation (inspection of equivalency) of the existing function of a software program to realize the reduction in the work for test (including a manual work) and the improvement in quality.

**SOLUTION:** This inspecting method for software program equivalency includes the test result extracting step of extracting prescribed information necessary for inspection of equivalency from the test result obtained by executing the software program before change and the software program after change on a computer system under the same test condition; and the collation step of collating the extraction information extracted from the test result of the software program before change obtained in the test result extracting step as a reference with the extraction information extracted from the test result of the software program after change for every prescribed test item.



## LEGAL STATUS

[Date of request for examination] 22.03.2001

[Date of sending the examiner's decision of rejection] 24.06.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision  
of rejection]

[Date of extinction of right]

**THIS PAGE BLANK (USPTO)**

## \* ,NOTICES \*

**JPO and NCIPJ are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

CLAIMS

---

[Claim(s)]

[Claim 1] In the software program equivalency verification approach of verifying the equivalency of the software program before modification, and the software program after modification on a computer system The test-result extract phase of extracting predetermined information required for verification of equivalency from the test result which performed the software program before said modification, and the software program after modification under the same test condition, It is based on the extract information extracted from the test result of the software program before said modification obtained in said test-result extract phase. The software program equivalency verification approach characterized by including the collating phase collated with the extract information extracted from the test result of the software program after said modification for every predetermined trial item.

[Claim 2] The extract information extracted from the test result of the software program after said modification said collating phase for said every predetermined trial item As opposed to the extract information extracted from the test result of the software program after said modification The software program equivalency verification approach according to claim 1 characterized by including the phase of judging those with equivalency (O.K.) when it is in predetermined tolerance, and judging those without equivalency (NG) when there is nothing into predetermined tolerance.

[Claim 3] Said software program equivalency verification approach is the software program equivalency verification approach according to claim 2 characterized by including further the collating result generation phase which generates a collating result file including the information as a result of the judgment in said collating phase for said every predetermined trial item.

[Claim 4] Said software program equivalency verification approach is the software program equivalency verification approach according to claim 3 characterized by including further the test-result management phase which generates the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for said every predetermined trial item based on said collating result file.

[Claim 5] In the software program equivalency verification equipment which verifies the equivalency of the software program before modification, and the software program after modification A test-result extract means to extract predetermined information required for verification of equivalency from the test result which performed the software program before said modification, and the software program after modification under the same test condition, It is based on the extract information extracted from the test result of the software program before said modification obtained with said test-result extract means. Software program equivalency verification equipment characterized by providing a collating means to collate with the extract information extracted from the test result of the software program after said modification, for every predetermined trial item.

[Claim 6] The extract information extracted from the test result of the software program after said modification said collating means for said every predetermined trial item As opposed to the extract information extracted from the test result of the software program after said modification The software program equivalency verification approach according to claim 5 characterized by judging those with equivalency (O.K.) when it is in predetermined tolerance, and judging those without equivalency (NG) further when there is nothing into predetermined tolerance.

[Claim 7] Said software program equivalency verification equipment is the software program equivalency verification approach according to claim 6 characterized by including further a collating result generation

means to generate a collating result file including the information as a result of the judgment in said collating means for said every predetermined trial item.

[Claim 8] Said software program equivalency verification equipment is the software program equivalency verification approach according to claim 7 characterized by including further the test-result management tool which generates the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for said every predetermined trial item based on said collating result file.

[Claim 9] The record medium with which the software with which the equipment turns into equipment which performs an approach according to claim 1 to 4 by installing in a computer apparatus was recorded and in which computer reading is possible.

---

[Translation done.]

\*,NOTICES \*

**JPO and NCIP are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the approach and equipment which verify the equivalency of the function modification before of the software program concerned, and after modification, when performing correction and making a change of a software program.

[0002]

[Description of the Prior Art] In recent years, various systems incorporating a software program are used. Although there is no request of a functional addition, when a program is changed in consideration of future maintenance nature to such a system, in order to check having not affected it out of a modification individual reason although the program was changed by functional addition like a rig RESSHON test (degradation trial), it is surely needed in the verification test of the existing function.

[0003] On the other hand, in large systems (telephone switchboard etc.), maintenance and improvement / functional addition of a system etc. is repeatedly performed over a long period. Moreover, even if small-scale, version up is made in many cases based on the resource with which the operating system and the application program were created in the past. For this reason, the program is still more complicated, when the design technique of the program included in the system and the structure of software may change with a creation time term or implementers and there is a functional addition. By current, by arranging a program, development supporting environment is unified and reexamination of software structure is performed for the purpose of shortening a design and test time from these things.

[0004] As mentioned above, when the correction and functional addition of a software program which were included in the system are performed, the verification test of the existing function is performed. The normality of the function of a software program was checked by examining based on the test procedure manual and the data for a trial which were carried out in the past, and conventionally, comparing this test result with the past test result (refer to drawing 3 (a)).

[0005]

[Problem(s) to be Solved by the Invention] However, the past test result needed to be managed, and grasp of detailed requirement specification was needed for the check of a test result, and such an approach had taken most time amount.

[0006] This invention was made in view of the above-mentioned point, can check the existing function of a software program automatically (verification of equivalency), and offers the software program equivalency verification approach and equipment which realize reduction of the activities for a trial (a human activity is included), and improvement in quality.

[0007]

[Means for Solving the Problem] The software program equivalency verification approach of this invention In the software program equivalency verification approach of verifying the equivalency of the software program before modification, and the software program after modification on a computer system The test-result extract phase of extracting predetermined information required for verification of equivalency from the test result which performed the software program before said modification, and the software program after modification under the same test condition, It is characterized by including the collating phase collated with the extract information extracted from the test result of the software program after said modification for every predetermined trial item on the basis of the extract information extracted from the test result of the software program before said modification obtained in said test-result extract phase.

[0008] In the software program equivalency verification approach of this invention moreover, said collating phase The extract information extracted from the test result of the software program after said modification for said every predetermined trial item It is characterized by including the phase of judging those with equivalency (O.K.) when it is in predetermined tolerance to the extract information extracted from the test result of the software program after said modification, and judging those without equivalency (NG) when there is nothing into predetermined tolerance.

[0009] Moreover, in the software program equivalency verification approach of this invention, it is characterized by including further the collating result generation phase which generates a collating result file including the information as a result of the judgment in said collating phase for said every predetermined trial item.

[0010] Said software program equivalency verification approach is characterized by including further the test-result management phase which generates the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for said every predetermined trial item based on said collating result file.

[0011] The software program equivalency verification equipment of this invention In the software program equivalency verification equipment which verifies the equivalency of the software program before modification, and the software program after modification A test-result extract means to extract predetermined information required for verification of equivalency from the test result which performed the software program before said modification, and the software program after modification under the same test condition, It is based on the extract information extracted from the test result of the software program before said modification obtained with said test-result extract means. It is characterized by providing a collating means to collate with the extract information extracted from the test result of the software program after said modification, for every predetermined trial item.

[0012] In the software program equivalency verification equipment of this invention moreover, said collating means The extract information extracted from the test result of the software program after said modification for said every predetermined trial item It is characterized by judging those with equivalency (O.K.), when it is in predetermined tolerance to the extract information extracted from the test result of the software program after said modification, and judging those without equivalency (NG) further, when there is nothing into predetermined tolerance.

[0013] Moreover, the software program equivalency verification equipment of this invention is characterized by including further a collating result generation means to generate a collating result file including the information as a result of the judgment in said collating means for said every predetermined trial item.

[0014] Moreover, the software program equivalency verification equipment of this invention is characterized by including further the test-result management tool which generates the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for said every predetermined trial item based on said collating result file.

[0015] Moreover, this invention is a record medium with which the software with which the equipment turns into equipment which performs an approach according to claim 1 to 4 was recorded and in which computer reading is possible by installing in a computer apparatus.

[0016]  
[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained with reference to a drawing. Drawing 1 is the block diagram showing the configuration of the equivalency verification equipment 1 which is the gestalt of 1 operation of the equipment which installed in the computer system the software program which consists of a procedure which is the gestalt of 1 operation of the software program equivalency verification approach of this invention, or the software program equivalency verification equipment of this invention.

[0017] The outline of the software program equivalency verification approach of this invention is explained first. The software program equivalency verification approach of this invention uses that the same result will be obtained if it examines on the same conditions to the software program before and behind modification since there is no modification in the existing function (function in which it does not change), when performing modification and a functional addition of a software program. That is, if the same test condition A as the software program before and behind modification is inputted as shown in drawing 2 , the test result A of the software program before modification and the test result B of the software program after modification will use

bringing the same result, will perform collating with a test result B by making a test result A into a reference value, will judge whether tolerance has a difference, and will verify equivalency of a software program.

[0018] Contrast with the approach ( drawing 3 (b) ) of verification of the equivalency of the software program before modification and after modification enforced in equivalency verification equipment 1 by drawing 3 and the conventional approach ( drawing 3 (a) ) is shown. With equivalency verification equipment 1, in order to examine on the same conditions to the software program before and behind modification, grasp of detailed requirement specification becomes unnecessary by there being no need of managing the past test result, and collating a test result automatically. In addition, although the number of trial items for examining by the software program before and behind modification in verifying of the equivalency of the software program before modification and after modification doubles compared with the conventional approach, about this, an automatic trial is possible and the load of a big activity is not applied.

[0019] With the gestalt of this operation explained below, since the information (response message etc.) from various testing-machine machines, the test atmosphere (workstation), etc. other than the message from a system is included in the actual test results A and B, it becomes difficult to compare the test result A used as criteria with a test result B simply. Then, the keyword appointed separately extracts data required for collating, and actual collating is performed using this extracted data (refer to drawing 4 ).

[0020] Here, the configuration of equivalency verification equipment 1 is explained. Equivalency verification equipment 1 consists of the storage section 2 which memorizes the test result A of the software program before modification, and the test result B of the software program after modification, the following test-result extract section 3a, collating section 3b, collating result generation section 3c, and the processing section 3 that consists of 3d of the test-result Management Department.

[0021] Test-result extract section 3a: From the test result A of the software program before modification, and the test result B of the software program after modification, extract based on the keyword which can define predetermined information required for verification of equivalency separately, and generate the file which consists of extracted information.

collating -- the section -- three -- b -- : -- a test result -- an extract -- the section -- three -- a -- a test result -- A -- from -- extracting -- having had -- an extract -- information -- A -- ' -- criteria -- \*\* -- carrying out -- the same -- a test result -- B -- from -- extracting -- having had -- an extract -- information -- B -- ' -- collating . At this time, it collates about the checkpoint appointed separately. And when extract information B' judges whether it is in predetermined tolerance from extract information A' used as criteria and is in predetermined tolerance, those with equivalency (O.K.) are judged, and when there is nothing into predetermined tolerance, those without equivalency (NG) are judged.

[0022] Collating result generation section 3c: Generate a collating result file including the information as a result of the judgment by collating section 3b for every trial item.

3d of test-result Management Department: Generate the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for every trial item based on the collating result file which collating result generation section 3c generated.

[0023] In addition, the above-mentioned test result A and a test result B are the results of performing each of the software program before modification, and the software program after modification, and examining on the basis of the test condition based on the trial item defined separately, and the system of a target. Moreover, it changes with systems so that naturally [ predetermined tolerance ], and it defines suitably for every system to examine.

[0024] Moreover, the storage section 2 is constituted by the recording device of the non-volatile of a hard disk, a magneto-optic disk, etc. in equivalency verification equipment 1. Moreover, the processing section 3 shall be constituted by memory, CPU (central processing unit), etc., and the function shall be realized by loading the software program (not shown) for realizing each function of the processing section 3 to memory, and performing it. Moreover, an input unit, a display, etc. shall be connected to equivalency verification equipment 1 as a peripheral device (neither is illustrated). Here, an input device means input devices, such as a keyboard and a mouse. A display means things, such as CRT (Cathode Ray Tube) and a liquid crystal display.

[0025] Next, actuation of the equivalency verification equipment 1 constituted in this way is explained.

[0026] First, based on a predetermined test procedure manual, the data for a trial predetermined in the system top of a target shall be used, the software program before modification and the software program after

modification shall be performed, the test result A of the software program before modification and the software program test result B after modification shall be obtained, and the examiner shall memorize at the storage section 2 of equivalency verification equipment 1.

[0027] The telephone switchboard should be used as the target especially in the following explanation. In this case, as an example of the file of the above-mentioned test result, a test-result file, a traffic data file, and a signal monitoring analysis file are raised. The example of contents of each above-mentioned file is shown in drawing 5 -7.

[0028] First, from a test-result file, test-result extract section 3a extracts predetermined information required for verification of equivalency based on keywords, such as "test initiation" and "normal termination", and as a result of consisting of such information, it generates a file (refer to drawing 9 ). Similarly, the data file and monitor file which consist of predetermined information to each are generated [ data file / traffic ] from a signal monitoring analysis file by making "employment information" etc. into a keyword, using "MSGTYPE" etc. as a keyword ( drawing 8 : step S1).

[0029] Next, in step S1, as a result of consisting of information which extracted predetermined information based on the keyword, from a file, a data file, and a monitor file, collating section 3b collates about the checkpoint shown in drawing 10 , and performs those with equivalency / nothing judgment ( drawing 8 : step S2).

[0030] In the example shown in drawing 10 , two results obtained from the test result of the software program after modification before modification, as a checkpoint to a file, it uses "false call normal call origination", "he having no urgent / the maximum emergency message output", and "he having no relief of various resources", OK/NG is judged, and collating and a judgment are further performed by making an "independence message", "various data output information", and "call-processing trace information" into a checkpoint. Similarly, to a data file, collating and a judgment are performed by making into a checkpoint "employment information" and "traffic information (the time information of an incompleted call (call which was not able to be connected normally), a distant-office call identity number, and a local station identification number outside of an object)." And to a monitor file, collating and a judgment are performed again by making into a checkpoint "signal monitoring information (the speech path acknowledge signal, the distant-office call identity number, and the local station identification number outside of an object)" which is the contents of a signal which a telephone switchboard receives.

[0031] In addition, although usually accessed by sending from telephone with the "false normal call origination" of a checkpoint to a telephone switchboard, since can prepare all the telephones to send and they cannot be examined, it is a checkpoint for checking that it has sent from false equipment normally (call origination) from creating the signal of various conditions with the false equipment equivalent to telephone, making it access to the exchange, and performing a verification test.

[0032] Moreover, since it is thought that there is a certain cause when the notice to a maintenance man when fault usually occurs to the condition of a telephone switchboard or connection of a telephone, saying a checkpoint "has no urgent / the maximum emergency message output" is and this message occurs at the time of test implementation, it is a checkpoint for this check. Moreover, although the resource which the telephone switchboard was usually performing resource management, such as various memory, saying a checkpoint "has no relief of various resources", caught the resource when the software program was performed normally, and was caught after processing termination is released, it is a checkpoint for checking whether a supplement and disconnection of this resource are made normally.

[0033] Moreover, if the "independence message" of a checkpoint has some of which the maintenance man other than urgent / the maximum emergency message is notified as a usual message, for example, time amount assignment registration of a certain command is carried out, when it becomes applicable time amount, it will be a message which notifies activation of a registration command. Moreover, the "call-processing trace information" of a checkpoint is the information concerning the command (call-processing trace command) for checking the normality of telephone connection. The event number (EEV-NO) which is the response from the telephone switchboard to the command for checking the normality of telephone connection, a connection first-move number, an addresser number, etc. are contained in this "call-processing trace information."

[0034] Next, collating result generation section 3c generates the file (collating result file) which includes a collating result for the result of collating and the judgment by collating section 3b for every trial item ( drawing 8 : step S3). To drawing 11 , an example of a collating result file of the case at the time (those with



equivalency) (with no equivalency) of Collating O.K. and Collating NG is illustrated, respectively. The date etc. is contained in these files the collating result of trace information besides a trial item number, emergency message information, and monitor information and an examiner name, and as a result of a judgment (OK/NG). The command parameter supplied to the telephone switchboard as a collating result of trace information at the time of Collating NG: The error message (ERR-XX generating) to RTR (the abbreviation for route trace) is recorded. In addition, emergency message information and monitor information are recorded as it is simply discriminable.

[0035] Next, 3d of test-result Management Department generates the test-result managed table of all trial items from each collating result file which collating result generation section 3c generated ( drawing 8 : step S4). An example of a test-result managed table is shown in drawing 12 . This test-result managed table consists of an item of an enforcing date the exception of the name of a trial item number and a test implementation person, EEV-NO (the above-mentioned event number), and OK/NG. The trial status-control person of the software program after modification can use this test-result managed table, and can perform a status control.

[0036] In addition, the flow of operation explained above is an example, and is not limited to the flow of the above-mentioned processing. Moreover, the software program which recorded on the record medium which can computer read the software program for realizing the function of the processing section 3 in drawing 1 , and was recorded on this record medium may be made to read into a computer system, and the software program equivalency verification approach may be enforced by performing. In addition, hardware, such as OS and a peripheral device, shall be included with a "computer system" here.

[0037] Moreover, "the record medium in which computer reading is possible" means storage, such as a hard disk built in portable media, such as a floppy (trademark) disk, a magneto-optic disk, ROM, and CD-ROM, and a computer system. Furthermore, the thing holding a fixed time amount software program shall also be included [ "whose record medium in which computer reading is possible" is ] like the volatile memory (RAM) inside the computer system used as a server when a software program is transmitted through communication lines, such as networks, such as the Internet, and the telephone line, or a client.

[0038] Moreover, the above-mentioned software program may be transmitted to other computer systems through a transmission medium from the computer system which stored this software program in storage etc. by the carrier wave in a transmission medium. Here, the "transmission medium" which transmits a software program says the thing of a medium which has the function to transmit information like communication lines (communication wire), such as networks (communication network), such as the Internet, and the telephone line. Moreover, the above-mentioned software program may be for realizing a part of function mentioned above. Furthermore, you may be what can realize the function mentioned above in combination with the software program already recorded on the computer system, and the so-called patch file (difference program).

[0039] As mentioned above, although the gestalt of implementation of this invention has been explained in full detail with reference to a drawing, a concrete configuration is not restricted to this operation gestalt, and the design of the range which does not deviate from the summary of this invention etc. is included.

[0040]  
[Effect of the Invention] As explained to the detail, according to this invention, as mentioned above, on a computer system The software program before modification, and the software program after modification Extract predetermined information required for verification of equivalency from the test result performed under the same test condition, and it is based on the extract information extracted from the test result of the software program before said modification obtained in said test-result extract phase. The extract information extracted from the test result of the software program after said modification is collated for every predetermined trial item. Thereby, in order to check the existing function of a software program (verification of the equivalency of the software program before and behind modification), it becomes unnecessary to manage the past test result, and as for an examiner, grasp of detailed requirement specification also becomes unnecessary, and the time amount for it does not need him, either. That is, reduction of the activities for a trial and improvement in quality are realized by the autocorrelation of a test result.

---

[Translation done.]

\*.NOTICES \*

**JPO and NCIP are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

## TECHNICAL FIELD

---

[Field of the Invention] This invention relates to the approach and equipment which verify the equivalency of the function modification before of the software program concerned, and after modification, when performing correction and making a change of a software program.

---

[Translation done.]

## \*.NOTICES \*

**JPO and NCIP are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

PRIOR ART

---

[Description of the Prior Art] In recent years, various systems incorporating a software program are used. Although there is no request of a functional addition, when a program is changed in consideration of future maintenance nature to such a system, in order to check having not affected it out of a modification individual reason although the program was changed by functional addition like a rig RESSHON test (degradation trial), it is surely needed in the verification test of the existing function.

[0003] On the other hand, in large systems (telephone switchboard etc.), maintenance and improvement / functional addition of a system etc. is repeatedly performed over a long period. Moreover, even if small-scale, version up is made in many cases based on the resource with which the operating system and the application program were created in the past. For this reason, the program is still more complicated, when the design technique of the program included in the system and the structure of software may change with a creation time term or implementers and there is a functional addition. By current, by arranging a program, development supporting environment is unified and reexamination of software structure is performed for the purpose of shortening a design and test time from these things.

[0004] As mentioned above, when the correction and functional addition of a software program which were included in the system are performed, the verification test of the existing function is performed. The normality of the function of a software program was checked by examining based on the test procedure manual and the data for a trial which were carried out in the past, and conventionally, comparing this test result with the past test result (refer to drawing 3 (a)).

---

[Translation done.]

\*,NOTIQUES \*

**JPO and NCIPJ are not responsible for any  
damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

## EFFECT OF THE INVENTION

---

[Effect of the Invention] As explained to the detail, according to this invention, as mentioned above, on a computer system The software program before modification, and the software program after modification Extract predetermined information required for verification of equivalency from the test result performed under the same test condition, and it is based on the extract information extracted from the test result of the software program before said modification obtained in said test-result extract phase. The extract information extracted from the test result of the software program after said modification is collated for every predetermined trial item. Thereby, in order to check the existing function of a software program (verification of the equivalency of the software program before and behind modification), it becomes unnecessary to manage the past test result, and as for an examiner, grasp of detailed requirement specification also becomes unnecessary, and the time amount for it does not need him, either. That is, reduction of the activities for a trial and improvement in quality are realized by the autocorrelation of a test result.

---

[Translation done.]

\*.NOTICES \*

**JPO and NCIPi are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

## TECHNICAL PROBLEM

---

[Problem(s) to be Solved by the Invention] However, the past test result needed to be managed, and grasp of detailed requirement specification was needed for the check of a test result, and such an approach had taken most time amount.

[0006] This invention was made in view of the above-mentioned point, can check the existing function of a software program automatically (verification of equivalency), and offers the software program equivalency verification approach and equipment which realize reduction of the activities for a trial (a human activity is included), and improvement in quality.

---

[Translation done.]

\*.NOTICES \*

**JPO and NCIP are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

## MEANS

---

[Means for Solving the Problem] The software program equivalency verification approach of this invention In the software program equivalency verification approach of verifying the equivalency of the software program before modification, and the software program after modification on a computer system The test-result extract phase of extracting predetermined information required for verification of equivalency from the test result which performed the software program before said modification, and the software program after modification under the same test condition, It is characterized by including the collating phase collated with the extract information extracted from the test result of the software program after said modification for every predetermined trial item on the basis of the extract information extracted from the test result of the software program before said modification obtained in said test-result extract phase.

[0008] In the software program equivalency verification approach of this invention moreover, said collating phase The extract information extracted from the test result of the software program after said modification for said every predetermined trial item It is characterized by including the phase of judging those with equivalency (O.K.) when it is in predetermined tolerance to the extract information extracted from the test result of the software program after said modification, and judging those without equivalency (NG) when there is nothing into predetermined tolerance.

[0009] Moreover, in the software program equivalency verification approach of this invention, it is characterized by including further the collating result generation phase which generates a collating result file including the information as a result of the judgment in said collating phase for said every predetermined trial item.

[0010] Said software program equivalency verification approach is characterized by including further the test-result management phase which generates the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for said every predetermined trial item based on said collating result file.

[0011] The software program equivalency verification equipment of this invention In the software program equivalency verification equipment which verifies the equivalency of the software program before modification, and the software program after modification A test-result extract means to extract predetermined information required for verification of equivalency from the test result which performed the software program before said modification, and the software program after modification under the same test condition, It is based on the extract information extracted from the test result of the software program before said modification obtained with said test-result extract means. It is characterized by providing a collating means to collate with the extract information extracted from the test result of the software program after said modification, for every predetermined trial item.

[0012] In the software program equivalency verification equipment of this invention moreover, said collating means The extract information extracted from the test result of the software program after said modification for said every predetermined trial item It is characterized by judging those with equivalency (O.K.), when it is in predetermined tolerance to the extract information extracted from the test result of the software program after said modification, and judging those without equivalency (NG) further, when there is nothing into predetermined tolerance.

[0013] Moreover, the software program equivalency verification equipment of this invention is characterized by including further a collating result generation means to generate a collating result file including the information as a result of the judgment in said collating means for said every predetermined trial item.

[0014] Moreover, the software program equivalency verification equipment of this invention is characterized by

including further the test-result management tool which generates the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for said every predetermined trial item based on said collating result file.

[0015] Moreover, this invention is a record medium with which the software with which the equipment turns into equipment which performs an approach according to claim 1 to 4 was recorded and in which computer reading is possible by installing in a computer apparatus.

[0016]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained with reference to a drawing. Drawing 1 is the block diagram showing the configuration of the equivalency verification equipment 1 which is the gestalt of 1 operation of the equipment which installed in the computer system the software program which consists of a procedure which is the gestalt of 1 operation of the software program equivalency verification approach of this invention, or the software program equivalency verification equipment of this invention.

[0017] The outline of the software program equivalency verification approach of this invention is explained first. The software program equivalency verification approach of this invention uses that the same result will be obtained if it examines on the same conditions to the software program before and behind modification since there is no modification in the existing function (function in which it does not change), when performing modification and a functional addition of a software program. That is, if the same test condition A as the software program before and behind modification is inputted as shown in drawing 2, the test result A of the software program before modification and the test result B of the software program after modification will use bringing the same result, will perform collating with a test result B by making a test result A into a reference value, will judge whether tolerance has a difference, and will verify equivalency of a software program.

[0018] Contrast with the approach ( drawing 3 (b)) of verification of the equivalency of the software program before modification and after modification enforced in equivalency verification equipment 1 by drawing 3 and the conventional approach ( drawing 3 (a)) is shown. With equivalency verification equipment 1, in order to examine on the same conditions to the software program before and behind modification, grasp of detailed requirement specification becomes unnecessary by there being no need of managing the past test result, and collating a test result automatically. In addition, although the number of trial items for examining by the software program before and behind modification in verifying of the equivalency of the software program before modification and after modification doubles compared with the conventional approach, about this, an automatic trial is possible and the load of a big activity is not applied.

[0019] With the gestalt of this operation explained below, since the information (response message etc.) from various testing-machine machines, the test atmosphere (workstation), etc. other than the message from a system is included in the actual test results A and B, it becomes difficult to compare the test result A used as criteria with a test result B simply. Then, the keyword appointed separately extracts data required for collating, and actual collating is performed using this extracted data (refer to drawing 4 ).

[0020] Here, the configuration of equivalency verification equipment 1 is explained. Equivalency verification equipment 1 consists of the storage section 2 which memorizes the test result A of the software program before modification, and the test result B of the software program after modification, the following test-result extract section 3a, collating section 3b, collating result generation section 3c, and the processing section 3 that consists of 3d of the test-result Management Department.

[0021] Test-result extract section 3a: From the test result A of the software program before modification, and the test result B of the software program after modification, extract based on the keyword which can define predetermined information required for verification of equivalency separately, and generate the file which consists of extracted information.

collating -- the section -- three -- b -- : -- a test result -- an extract -- the section -- three -- a -- a test result -- A -- from -- extracting -- having had -- an extract -- information -- A -- ' -- criteria -- \*\* -- - carrying out -- the same -- a test result -- B -- from -- extracting -- having had -- an extract -- information -- B -- ' -- collating . At this time, it collates about the checkpoint appointed separately. And when extract information B' judges whether it is in predetermined tolerance from extract information A' used as criteria and is in predetermined tolerance, those with equivalency (O.K.) are judged, and when there is nothing into predetermined tolerance, those without equivalency (NG) are judged.

[0022] Collating result generation section 3c: Generate a collating result file including the information as a

result of the judgment by collating section 3b for every trial item.

3d of test-result Management Department: Generate the test-result managed table which recorded information including the quality (OK/NG) of the test result of all trial items for every trial item based on the collating result file which collating result generation section 3c generated.

[0023] In addition, the above-mentioned test result A and a test result B are the results of performing each of the software program before modification, and the software program after modification, and examining on the basis of the test condition based on the trial item defined separately, and the system of a target. Moreover, it changes with systems so that naturally [ predetermined tolerance ], and it defines suitably for every system to examine.

[0024] Moreover, the storage section 2 is constituted by the recording device of the non-volatile of a hard disk, a magneto-optic disk, etc. in equivalency verification equipment 1. Moreover, the processing section 3 shall be constituted by memory, CPU (central processing unit), etc., and the function shall be realized by loading the software program (not shown) for realizing each function of the processing section 3 to memory, and performing it. Moreover, an input unit, a display, etc. shall be connected to equivalency verification equipment 1 as a peripheral device (neither is illustrated). Here, an input device means input devices, such as a keyboard and a mouse. A display means things, such as CRT (Cathode Ray Tube) and a liquid crystal display.

[0025] Next, actuation of the equivalency verification equipment 1 constituted in this way is explained.

[0026] First, based on a predetermined test procedure manual, the data for a trial predetermined in the system top of a target shall be used, the software program before modification and the software program after modification shall be performed, the test result A of the software program before modification and the software program test result B after modification shall be obtained, and the examiner shall memorize at the storage section 2 of equivalency verification equipment 1.

[0027] The telephone switchboard should be used as the target especially in the following explanation. In this case, as an example of the file of the above-mentioned test result, a test-result file, a traffic data file, and a signal monitoring analysis file are raised. The example of contents of each above-mentioned file is shown in drawing 5 -7.

[0028] First, from a test-result file, test-result extract section 3a extracts predetermined information required for verification of equivalency based on keywords, such as "test initiation" and "normal termination", and as a result of consisting of such information, it generates a file (refer to drawing 9 ). Similarly, the data file and monitor file which consist of predetermined information to each are generated [ data file / traffic ] from a signal monitoring analysis file by making "employment information" etc. into a keyword, using "MSGTYPE" etc. as a keyword ( drawing 8 : step S1).

[0029] Next, in step S1, as a result of consisting of information which extracted predetermined information based on the keyword, from a file, a data file, and a monitor file, collating section 3b collates about the checkpoint shown in drawing 10 , and performs those with equivalency / nothing judgment ( drawing 8 : step S2).

[0030] In the example shown in drawing 10 , two results obtained from the test result of the software program after modification before modification, as a checkpoint to a file, it uses "false call normal call origination", "he having no urgent / the maximum emergency message output", and "he having no relief of various resources", OK/NG is judged, and collating and a judgment are further performed by making an "independence message", "various data output information", and "call-processing trace information" into a checkpoint. Similarly, to a data file, collating and a judgment are performed by making into a checkpoint "employment information" and "traffic information (the time information of an incompleted call (call which was not able to be connected normally), a distant-office call identity number, and a local station identification number outside of an object)." And to a monitor file, collating and a judgment are performed again by making into a checkpoint "signal monitoring information (the speech path acknowledge signal, the distant-office call identity number, and the local station identification number outside of an object)" which is the contents of a signal which a telephone switchboard receives.

[0031] In addition, although usually accessed by sending from telephone with the "false normal call origination" of a checkpoint to a telephone switchboard, since can prepare all the telephones to send and they cannot be examined, it is a checkpoint for checking that it has sent from false equipment normally (call origination) from creating the signal of various conditions with the false equipment equivalent to telephone, making it access to



the exchange, and performing a verification test.

[0032] Moreover, since it is thought that there is a certain cause when the notice to a maintenance man when fault usually occurs to the condition of a telephone switchboard or connection of a telephone, saying a checkpoint "has no urgent / the maximum emergency message output" is and this message occurs at the time of test implementation, it is a checkpoint for this check. Moreover, although the resource which the telephone switchboard was usually performing resource management, such as various memory, saying a checkpoint "has no relief of various resources", caught the resource when the software program was performed normally, and was caught after processing termination is released, it is a checkpoint for checking whether a supplement and disconnection of this resource are made normally.

[0033] Moreover, if the "independence message" of a checkpoint has some of which the maintenance man other than urgent / the maximum emergency message is notified as a usual message, for example, time amount assignment registration of a certain command is carried out, when it becomes applicable time amount, it will be a message which notifies activation of a registration command. Moreover, the "call-processing trace information" of a checkpoint is the information concerning the command (call-processing trace command) for checking the normality of telephone connection. The event number (EEV-NO) which is the response from the telephone switchboard to the command for checking the normality of telephone connection, a connection first-move number, an addresser number, etc. are contained in this "call-processing trace information."

[0034] Next, collating result generation section 3c generates the file (collating result file) which includes a collating result for the result of collating and the judgment by collating section 3b for every trial item ( drawing 8 : step S3). To drawing 11 , an example of a collating result file of the case at the time (those with equivalency) (with no equivalency) of Collating O.K. and Collating NG is illustrated, respectively. The date etc. is contained in these files the collating result of trace information besides a trial item number, emergency message information, and monitor information and an examiner name, and as a result of a judgment (OK/NG). The command parameter supplied to the telephone switchboard as a collating result of trace information at the time of Collating NG: The error message (ERR-XX generating) to RTR (the abbreviation for route trace) is recorded. In addition, emergency message information and monitor information are recorded as it is simply discriminable.

[0035] Next, 3d of test-result Management Department generates the test-result managed table of all trial items from each collating result file which collating result generation section 3c generated ( drawing 8 : step S4). An example of a test-result managed table is shown in drawing 12 . This test-result managed table consists of an item of an enforcing date the exception of the name of a trial item number and a test implementation person, EEV-NO (the above-mentioned event number), and OK/NG. The trial status-control person of the software program after modification can use this test-result managed table, and can perform a status control.

[0036] In addition, the flow of operation explained above is an example, and is not limited to the flow of the above-mentioned processing. Moreover, the software program which recorded on the record medium which can computer read the software program for realizing the function of the processing section 3 in drawing 1 , and was recorded on this record medium may be made to read into a computer system, and the software program equivalency verification approach may be enforced by performing. In addition, hardware, such as OS and a peripheral device, shall be included with a "computer system" here.

[0037] Moreover, "the record medium in which computer reading is possible" means storage, such as a hard disk built in portable media, such as a floppy (trademark) disk, a magneto-optic disk, ROM, and CD-ROM, and a computer system. Furthermore, the thing holding a fixed time amount software program shall also be included [ "whose record medium in which computer reading is possible" is ] like the volatile memory (RAM) inside the computer system used as a server when a software program is transmitted through communication lines, such as networks, such as the Internet, and the telephone line, or a client.

[0038] Moreover, the above-mentioned software program may be transmitted to other computer systems through a transmission medium from the computer system which stored this software program in storage etc. by the carrier wave in a transmission medium. Here, the "transmission medium" which transmits a software program says the thing of a medium which has the function to transmit information like communication lines (communication wire), such as networks (communication network), such as the Internet, and the telephone line. Moreover, the above-mentioned software program may be for realizing a part of function mentioned above. Furthermore, you may be what can realize the function mentioned above in combination with the

software program already recorded on the computer system, and the so-called patch file (difference program).  
[0039] As mentioned above, although the gestalt of implementation of this invention has been explained in full detail with reference to a drawing, a concrete configuration is not restricted to this operation gestalt, and the design of the range which does not deviate from the summary of this invention etc. is included.

---

[Translation done.]

**\*.NOTICES \***

**JPO and NCIPi are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

**DESCRIPTION OF DRAWINGS**

---

**[Brief Description of the Drawings]**

**[Drawing 1]** It is the block diagram showing the configuration of the equivalency verification equipment which is the gestalt of 1 operation of this invention.

**[Drawing 2]** It is drawing explaining the software program equivalency verification approach of this invention.

**[Drawing 3]** It is drawing contrasting and showing the conventional approach and the software program equivalency verification approach of this invention.

**[Drawing 4]** It is drawing explaining the equivalency verification equipment which is the gestalt of 1 operation of this invention.

**[Drawing 5]** It is an example of a test-result file.

**[Drawing 6]** It is an example of a traffic data file.

**[Drawing 7]** It is an example of a signal monitoring analysis file.

**[Drawing 8]** It is the operation flow chart of the equivalency verification equipment which is the gestalt of 1 operation of this invention.

**[Drawing 9]** It is drawing showing an example of the extract file generated with the keyword applied for every test-result file.

**[Drawing 10]** It is drawing showing an example of the checkpoint for every extract file.

**[Drawing 11]** It is an example of a collating result file.

**[Drawing 12]** It is an example of a test-result managed table.

**[Description of Notations]**

1 --- Equivalency verification equipment 2 -- Storage section

3 --- Processing section 3a --- Test-result extract section

3b --- Collating section 3c --- Collating result generation section

3d --- Test-result Management Department

---

**[Translation done.]**

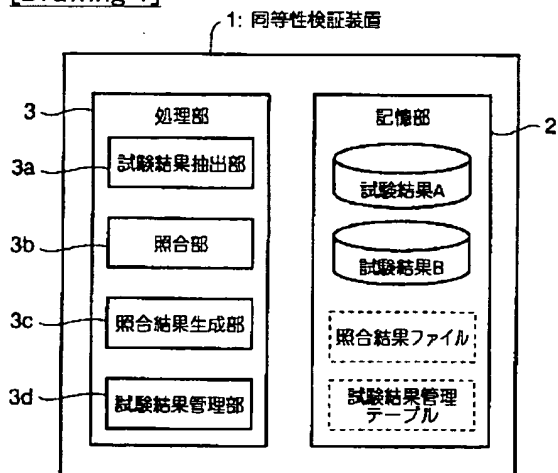
## \* NOTICES \*

JPO and NCIP are not responsible for any damages caused by the use of this translation.

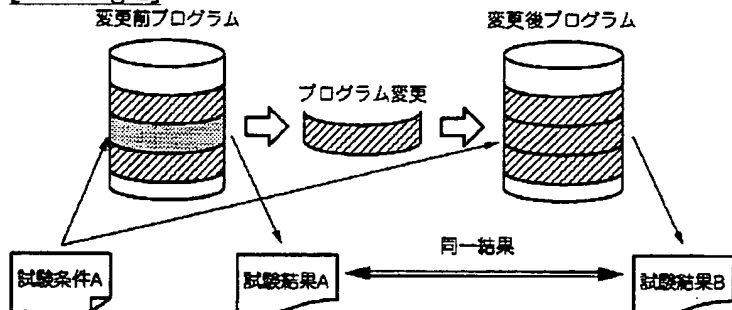
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

## DRAWINGS

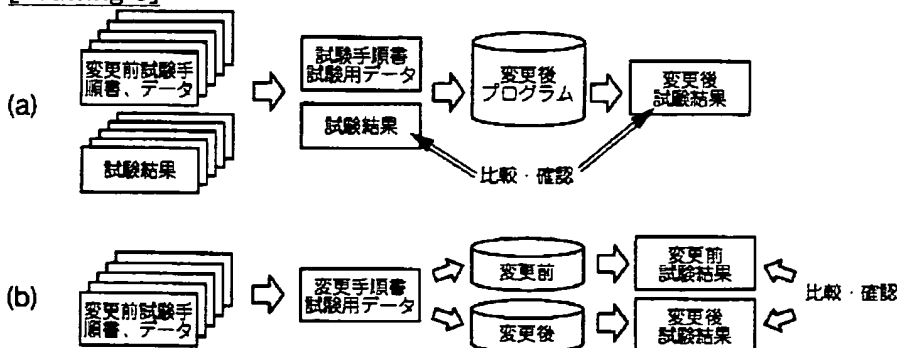
[Drawing 1]



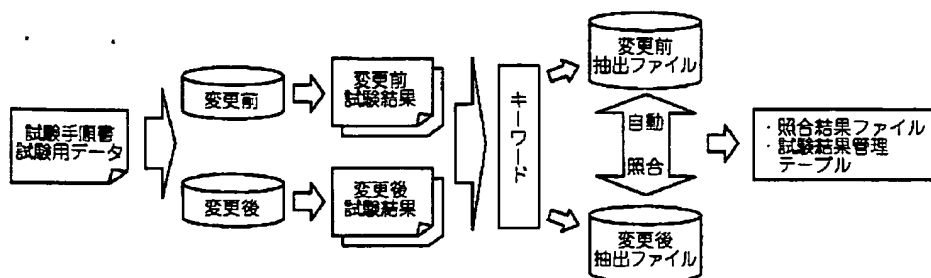
[Drawing 2]



[Drawing 3]



[Drawing 4]

**[Drawing 5]**

試験結果ファイル

### 試験開始: bua210001 ### ← キーワード  
 ¥DTY Y1999.M11.D26.H10.M10

### トラヒック測定 ###

¥DSP TRF/

# DSPTRF #

000000000 000000000 000000000

...DSP END 11/26 10:10 (A)

} チェックポイント

共通線信号ファイルを設定した

## 1 call 発呼 ##

CSR CNT=00000000 SI=ANET SSF=00 HiHo=SDRQ CiCo=00 ← チェックポイント

# ..SCP.TRACE START 11/26 10:10 (A) ← チェックポイント

SVCNID:ALP AN:990xxxxxx

### TRAFFIC DATA OUT (DBG) ###

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

¥DBG SVC OUT/

### SVC TRF:SVCTFT (DBG) ###

SETUZOKU FUKANOU : 000000001 UP

BUSY SOUGUU : 000000001 UP

} チェックポイント

**[Drawing 6]**

トラヒックデータファイル

★★★ 運用情報解析結果 (NSP共通) ★★★ ← キーワード

該当の試験では、運用情報の出力がありません

●●● トラヒック情報出力(ALP) Ver2.00 ●●●

世代識別 : [1]

データ識別 : [9]

ユーザ情報幅 : [60]

発信局番号 : [80-02-018]

情報提供事業者サービス : [990xxxxxx]

} チェックポイント

**[Drawing 7]**

信号モニタ解析ファイル

MSGTYPE DPC OPC SLS CLS ← キーワード

XUOT 1A5E 000E 00 81

データ長 : [37]

ユーザクラス : [1:高度電話網サービス]

相手局呼識別番号 : [000000]

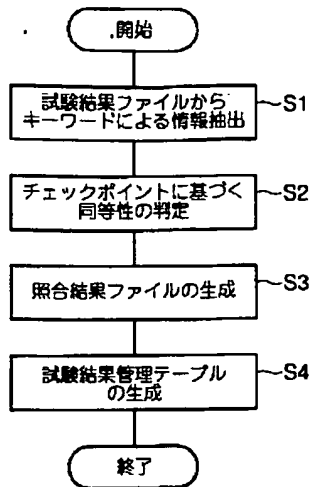
自局呼識別番号 : [201100]

シーケンス番号 : [1]

信号種別 : [11:加入者データ問合せ]

} チェックポイント

**[Drawing 8]**



[Drawing 9]

試験結果のファイル	キーワード	抽出ファイル
試験結果ファイル	試験開始、正常終了	結果ファイル
トラヒックデータファイル	運用情報	データファイル
信号モニタ解析ファイル	MSGTYPE	モニタファイル

[Drawing 10]

抽出ファイル	チェックポイント
結果ファイル	類似呼正常発呼 緊急/最緊急メッセージ出力無し 各種リソースの浮き上がり無し 自立メッセージ照合 各種データ出力情報照合 呼処理トレース情報照合
データファイル	運用情報、トラヒック情報照合 (不完了呼の時刻情報と相手局呼識別番号、 自局識別番号は照合対象外)
モニタファイル	信号モニタ情報照合 (通話路確認信号と相手局呼識別番号、 自局識別番号は照合対象外)

[Drawing 11]

照合OK時	照合NG時
>>>> 試験項目番号 >>>> —— トレース情報 —— —— 緊急メッセージ情報 —— —— モニタ情報 —— 試験項目番号 試験者名 OK 日付	>>>> 試験項目番号 >>>> —— トレース情報 —— RTR...ERR-XX発生 —— 緊急メッセージ情報 —— XXXX スタック 収集 NG —— モニタ情報 —— 結果 xxx 情報 0:必要 期待値 xxx 情報 1:不要 試験項目番号 試験者名 NG 日付

[Drawing 12]

試験項目番号	試験実施者	EEV-NO	OK/NG	実施日
0001	担当者名	00000000	OK	0209
0002	担当者名	00000000	OK	0209
0003	担当者名	00000000	OK	0209
0004	担当者名	03130408	NG	0209
⋮	⋮	⋮	⋮	⋮

[Translation done.]

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record.**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**